UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/824,251	04/13/2004	Andreas W. Wendker	APLE.P0004C	6431

62224   7590   10/15/2008
ADELI & TOLLEN, LLP
1875 CENTURY PARK EAST
SUITE 1360
LOS ANGELES, CA 90067

EXAMINER
MYINT, DENNIS Y

ART UNIT	PAPER NUMBER
2162	

MAIL DATE	DELIVERY MODE
10/15/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

	Application No.	Applicant(s)
Office Action Summary	10/824,251	WENDKER ET AL.
	Examiner	Art Unit
	DENNIS MYINT	2162

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on <u>16 June 2008</u>.
2a)☐ This action is **FINAL**.   2b)☒ This action is non-final.
3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) <u>20-27,31,32,35-38,42-44,46-53,56-63 and 75-88</u> is/are pending in the application.
    4a) Of the above claim(s) _____ is/are withdrawn from consideration.
5)☐ Claim(s) _____ is/are allowed.
6)☒ Claim(s) <u>20-27, 31-32, 35-38, 42-44, 46-53 and 56-63, and 75-88</u> is/are rejected.
7)☐ Claim(s) _____ is/are objected to.
8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.
10)☒ The drawing(s) filed on <u>13 April 2004</u> is/are: a)☐ accepted or b)☒ objected to by the Examiner.
    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
    a)☐ All   b)☐ Some * c)☐ None of:
        1.☐ Certified copies of the priority documents have been received.
        2.☐ Certified copies of the priority documents have been received in Application No. _____.
        3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
    Paper No(s)/Mail Date <u>07/25/2008</u>.
4) ☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Continued Examination Under 37 CFR 1.114

1.   A request for continued examination under 37 CFR 1.114, including the fee set

forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this

application is eligible for continued examination under 37 CFR 1.114, and the fee set

forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action

has been withdrawn pursuant to 37 CFR 1.114. Applicant's submissions filed on May

27, 2008, and June 16, 2008, have been entered.

2.   Claims 20-27, 31-32, 35-38, 42-44, 46-53 and 56-63, and 75-88 are currently

pending in this application. Claims 20, 31, 35, 42, 46, 52, 58, 59, 80, 84, and 87 are

independent claims. In the amendment filed on May 27, 2008, claims 20, 27, 35, 36, 42,

and 44 were amended and claims 64-74 were cancelled. In the supplemental

amendment filed on June 16, 2008, claims 59 and claims 77-88 are newly added.

3.   In light of the amendment made to claim 59, objection to the specification with

respect to claim 59 in the prior office action is hereby withdrawn.

Information Disclosure Statement

4.   The references cited in the information disclosure statement (IDS), submitted on

July 25, 2008, have been considered by the Examiner.

Response to Arguments

5.   Applicant's arguments filed on May 27, 2008, have been considered but are not persuasive.

With respect to rejection of claims 46-51 under 35 U.S.C. 101 in the prior office action, Applicant in the remarks argued that *"Applicants respectfully disagree that a system is directed towards software per se if it includes a database and a server"* (Applicant's remarks, Page 14 last paragraph and Page 16 third paragraph).

In response, it is pointed out that Claims 46-51 is directed to a system. Said system is software per se. Specification only defines "a computer system" – not "system". A *server* as defined in the abstract of the specification is an application server. Additionally, Figure 2 of the specification clearly defines a "a server communicatively coupled to the database", which is all software per se. Therefore, said system of claim 46-51 is software per se and does not fall within the four statutory categories.

Referring to claims 20-27, Applicant also argued that *"Galea does not disclose all claim limitations to anticipate claim 20. For instance, Galea does not disclose classifying entities into entity types after receiving a request from a client"* (Applicant's argument, page 15 second paragraph). Referring to claims 31-33, Applicant also argued that *"Galea does not disclose a computer program that has instructions for classifying entities into entity types'"* (Applicant's argument, page 16 last paragraph).

Examiner respectfully disagrees all of the allegations as argued. Examiner, in his previous office action, gave detail explanation of claimed limitation and pointed out exact locations in the cited prior art. Examiner is entitled to give claim limitations their broadest reasonable interpretation in light of the specification. See MPEP 2111 [R-1]

Interpretation of Claims-Broadest Reasonable Interpretation.

During patent examination, the pending claims must be 'given the broadest reasonable interpretation consistent with the specification.' Applicant always has the opportunity to amend the claims during prosecution and broad interpretation by the examiner reduces the possibility that the claim, once issued, will be interpreted more broadly than is justified. In re Prater, 162 USPQ 541,550-51 (CCPA 1969).

In response, it is respectfully pointed out that said argument is moot in view of the new ground(s) of rejection, that is, Galea in view of Cason. Galea in view of Cason (particularly Cason) teaches classifying entities into entity types as follows: **"after receiving said request, classifying the plurality of entities into entity types"** (Cason Column 4 Lines 29-57, i.e., *In an ERwin diagram, the entity is represented by drawing a box 280 with the name of the entity on top and all of the attributes of the entity inside the box. The horizontal line 281 in the entity box divides the attributes into two sets: keys and non-keys (or, data). The area above the line is the key area 282, and the area below the line is the data area 284.* ***The key is the set of attributes that identifies an entity****. A key attribute is an attribute which, either by itself, or in combination with other key attributes, forms a unique identifier for an entity. Primary keys are placed above the line in the key area. A non-key attribute is an attribute which has not been chosen as a key, and is placed in the data area.* ***Entity instances in an ERwin model are always identified by key attributes.*** *Key attributes are **chosen** from a set of candidate key attributes, which are attributes, or groups of attributes, which can be selected as **primary keys**, and as a candidate key must uniquely identify*

each instance of the entity, and may have no part null, i.e., empty or missing. In

*choosing a primary key from the set of candidate key attributes, **the data modeler trys***

to find an attribute which will not change its value over the life of each instance of

***the entity**, is reasonably small, and is not an intelligent key. An intelligent key is a key*

where the structure of the digits in the identifer indicates groupings, locations,

classifications, dates or the like. Candidate keys not selected as primary keys are

designated as alternate keys, and the symbol AKn, where n is a number, is placed after

*those attributes which form the **alternate key**. Alternate keys are used to show different*

indexes the business will use to access the data; Herein, **key attributes** of database

entities "map" to the "entities" of the claimed invention, which are classified as primary

keys, alternate keys and so on).

Referring to claim 35, Applicant argued that "*The office action rejected claim 35*

under the same rationale as claim 20. According, for reasons similar to those stated

above for claim 20, Applicants respectfully submit that Galea does not disclose sending

a request for a user interface, wherein the classification of entities into entity types is

initiated by the request" (Applicant's argument, page 17 last paragraph through page 18

first paragraph).

In response, it is respectfully pointed out that claim 35 is also rejected on the new

ground(s) of rejection, that is, Galea in view of Cason.

Referring to claims 42-44, Applicant argued that "*For instance, Galea does not*

disclose receiving a description of a user interface, where the description is based on

classification of entities into entity types" (Applicant's argument, page 19 first

paragraph).

In response, it is respectfully pointed out that claim 42 is also rejected on the new

ground(s) of rejection, that is, Galea in view of Cason.

Referring to claims 46-51, Applicant argued that "*Applicants respectfully submit*

that Galea does not disclose the system of claim 46. In the Office Action, the Examiner

rejected claims 46 under the same rationale as claim 42. Accordingly, for reasons

similar to those stated above for claim 42, Applicants respectfully submit that Galea

does not disclose creating a description, where the description is based on classification

of entities into entities" (Applicant's argument, page 20 second paragraph).

In response, it is respectfully pointed out that claim 46 is also rejected on the new

ground(s) of rejection, that is, Galea in view of Cason.

Referring to claims 52-57, Applicant argued that "*Applicants respectfully submit*

that Galea does not disclose the computer of claim 52. For instance, Galea does not

disclose a computer that includes an application for generating a user interface by using

and description and by retrieving a data set from a data store to populate at least one

user interface element. Instead, Galea describes" (Applicant's argument, page 21

second paragraph).

In response, it is pointed out that Galea in view of Cason teaches "retrieving a

data set from a data store to populate at least one user interface element" as follows:

Galea, Galea, Column 5 Lines 51-59 i.e., *A user accesses a server through client 106,*

108 via wide area network 112 in order to access items for sale (domains) in the

product and GUI database. Each product or service configuration domain, together with

the user interface tags, is compiled into a secure, binary compressed file format. After a

user accesses the domain through client 106, 108, the compiled configuration model is

downloaded to the browser of client 106*, 108. The user interface is dynamically*

generated based upon the compiled domain tags; ***Galea*** *Column 11 line 64 through*

Column 12 line 16, i.e., FIG. 9 illustrates an exemplary information block input window

900. Window 900 is displayed after a user selects a particular domain for viewing. The

information used to draw or paint to the HTML page are shown in FIG. 9 are

downloaded from server to client after a user accesses a particular domain. In the

example of FIG. 9, the information is used to ***display a medicine or drugstore e-***

commerce application*. After the user accesses a particular domain, the information is*

downloaded to the client from the server and the client applet downloaded from the

server is used to create the display. Window 900 includes a number of fields for

selecting specific information contained in the downloaded interactive decision map file.

FIG. 10 illustrates an exemplary interactive decision map window 1000. Decision map

window 1000 is displayed after the user selects specific information from information

block input window 800. ***Window 1000 includes a number of input and drop***

selection boxes and for choosing and selecting information downloaded from

server 102*).*

Referring to claims 58 and 63, Applicant argued that "

For at least the following reasons, Applicants respectfully submit that Galea does

not disclose the method of claim 58. For instance, the cited reference does not disclose

supplying a first description to transact with a first data store and a second description

to transact with a second data store. The Office Action cited Galea, column 5, lines 19-

22" (Applicant's argument, page 22 second paragraph).

In response, it is pointed out that Galea Column 5 Lines 51-55 teaches that "A

user access a server through client 106, 108 via wide area network 112 in order to

access items for sale (domains) in the product and GUI database. Each product or

service configuration domain, together with the user interfaces tags , is compiled into a

secure, binary compressed file format. And Galea, Column 5 Lines 10-22, i.e., FIG. 1 is

an illustration of one embodiment for a distributed client-side e-commerce service

system 100. Referring to FIG. 1, server 102 is coupled to mass storage device 104.

Server 102 and mass storage device 104 are coupled via wide area network (WAN) 112

to a variety of clients 106 and 108. Wide area network 112 may be coupled to any of a

variety of clients 106 and 108. In one embodiment, mass storage device 104 contains a

product component and graphical user interface (GUI) database and an interactive

decision map database. In an alternate embodiment, each of the databases may be

contained in a separate mass storage devices 104; Figure 12 and Figure 13; Column 12

Lines 28-50, which explains Figure 12 and Figure 13; Also note Figure 1 wherein Client

106 (first client) and Client 108 (second client) can access the server for different user

interfaces); Particular note Galea's disclosure which states that **"In an alternative**

embodiment, each of the databases may be contained in a separate mass storage

devices 104, which clearly teaches that there are more than one data stores which

different clients (users) could access. In addition, Galea, Column 4 Lines 1-5 that the

constraint-based configuration file is mapped into a **plurality of pages** at a client, and a

page of **the plurality of page** is updated directly from the constraint-based

configuration file. Galea Column 5 Lines 19-22 teaches that "*In an alternate*

embodiment, **each of the databases may be contained in a separate mass storage**

devices 104 and Column 5 Lines 23-26 teaches that "*In one embodiment, the*

databases contain compiled product or service components (domains). Each domain

contains descriptive information which bind the domain elements to their respective GUI

representations". Also note Figure 12 and Figure 13, Column 12 Lines 28-50, which

explains Figure 12 and Figure 13 and Also note Figure 1 **wherein Client 106 (first**

client) and Client 108 (second client) can access the server for different user

interfaces.

As per Applicant's argument regarding "different descriptions based on the

roles of the users" as per claims 59-62, said argument is moot because Applicant

has amended claim 59 discarding the limitation "roles of users".

In view of the above, the examiner contends that all limitations as recited in the

claims have been addressed in this Action. For the above reasons, Examiner believed

that rejection of the last Office action was proper.

Specification

6.   The specification is objected to as failing to provide proper antecedent basis for

the claimed subject matter. See 37 CFR 1.75(d)(1) and MPEP § 608.01(o). Correction

of the following is required: independent claims 31 and 42 recites "a computer readable

medium". However, the specification of the instant application fails to provide proper

antecedent basis for the limitation "computer readable medium".

Additionally, claim 82 recites "wherein said request for data to populate the user

interface element is received automatically **from the client** prior to receiving a

transaction or a query from the user of said client". However, the specification of the

instant application fails to provide proper antecedent basis for "automatically receiving

data from the client".

Drawings

7.   The drawings are objected to under 37 CFR 1.83(a). The drawings must show

every feature of the invention specified in the claims. Therefore, the limitation

"computer readable medium" as recited in claims 31 and 42 must be shown or the

feature(s) canceled from the claim(s). No new matter should be entered.

Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in

reply to the Office action to avoid abandonment of the application. Any amended

replacement drawing sheet should include all of the figures appearing on the immediate

prior version of the sheet, even if only one figure is being amended. The figure or figure

number of an amended drawing should not be labeled as "amended." If a drawing figure

is to be canceled, the appropriate figure must be removed from the replacement sheet,

and where necessary, the remaining figures must be renumbered and appropriate

changes made to the brief description of the several views of the drawings for

consistency. Additional replacement sheets may be necessary to show the renumbering

of the remaining figures. Each drawing sheet submitted after the filing date of an

application must be labeled in the top margin as either "Replacement Sheet" or "New

Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the examiner,

the applicant will be notified and informed of any required corrective action in the next

Office action. The objection to the drawings will not be held in abeyance.

Claim Rejections - 35 USC § 101

8.   35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

9.   Claims 31-32, 77-78, 42-43, 46-51, 52-53, 56-57, 58, 63, 59, 60, 62, and 80-82

are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-

statutory subject matter.

As per claims 31 and 42, the claim is directed to "computer-readable medium".

However, the specification of the claimed invention fails to define/limit "computer

readable medium", said computer readable medium is give the broadest, reasonable

interpretation to include wireless signals and wave, which do not fall into the four

statutory categories. As such, claims 31 and 42 are rejected under 35 U.S.C. 101 as

being directed to non-statutory subject matter.

Claims 32 and 77-78 depend on claim 31 and claims 43-44 depend on claim 42.

These are also rejected under 35 U.S.C. 101 by virtue of their dependency on claims 31

and 42.

Claims 46-51 is directed to a system. Said system is software per se. Specification only defines "a computer system" – not "system". A *server* as defined in the abstract of the specification is an application server. Additionally, Figure 2 of the specification clearly defines a "a server communicatively coupled to the database", which is all software per se. Therefore, said system of claim 46-51 is software per se and does not fall within the four statutory categories.

Applicant in the remarks argued that ""Claims 46-51 provide a tangible result through their non-abstract matter. The database must necessarily be non-abstract (i.e., tangible)" (Applicant's remarks, Page 15 last paragraph and Page 16 first paragraph). Applicant is respectfully reminded that claims 46-51 are rejected under 35 U.S.C. 101 as being software per se - not because the subject matter is abstract or not because the claim does not produce concrete results.

Claims 47-51 are rejected 35 U.S.C. 101 because said claims depend on claim 46.

Claim 52 is directed to a "computer". Said "computer" is given the broadest, reasonable interpretation to mean "a computer system". According to the claim language and in light of figures 2 and 3, said computer system is software per se computer system. As such, the claim is directed to non-statutory subject matter and is rejected under 35 U.S.C. 101.

Dependent claims of claim 52 (i.e., claims 53 and 56-57) are also rejected under 35 U.S.C. 101 by virtue of their dependency on claim 52.

As per claim 58, the claim recites "to generate the first user interface" (line 4) and "to generate the second user face" (line 7). As such, the claim is directed to intended use and is not actually "generating" a user interface. Therefore, the claim fails to transform underlying subject matter to a different state or thing and is rejected under 35 U.S.C. 101 as being directed to non-statutory subject matter.

Claim 63 depends on claim 58 and is also rejected under 35 U.S.C. 101 by virtue of its dependency on claim 63.

As per claim 59, the claim is directed to "receiving user authentication information" and fails to transform underlying subject matter to a different state or thing and is rejected under 35 U.S.C. 101 as being directed to non-statutory subject matter.

Claims 60 and 62 are rejected under 35 U.S.C. 101 by virtue of their dependency on claim 59.

As per claim 80, the claim recites "said client enabled to generate said user interface which permits a user of said client to transact with said data store" (line 10). As such, the claim is directed to intended use and is not actually "generating" a user interface. Therefore, the claim fails to transform underlying subject matter to a different state or thing and is rejected under 35 U.S.C. 101 as being directed to non-statutory subject matter.

Claims 81-82 depend on claim 80 and is also rejected under 35 U.S.C. 101 by virtue of its dependency on claim 80.

Claim Rejections - 35 USC § 103

10.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set
> forth in section 102 of this title, if the differences between the subject matter sought to be patented and
> the prior art are such that the subject matter as a whole would have been obvious at the time the
> invention was made to a person having ordinary skill in the art to which said subject matter pertains.
> Patentability shall not be negatived by the manner in which the invention was made.

11.   This application currently names joint inventors. In considering patentability of

the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of

the various claims was commonly owned at the time any inventions covered therein

were made absent any evidence to the contrary. Applicant is advised of the obligation

under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was

not commonly owned at the time a later invention was made in order for the examiner to

consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g)

prior art under 35 U.S.C. 103(a).

12.   Claim 20, 24-26, 31, 35-37, 42-44, 46-50, 52-53, 56-58, 63, and 75-88 are

rejected under 35 U.S.C. 103(a) as being unpatentable over Galea (hereinafter "Galea",

U.S. Patent Number 6404445) in view Cason (U.S. Patent Number 6035300).

        As per claim 20, Galea is directed to a method for creating a description of a user

interface that transacts with a database having a data model containing a plurality of

entities, the description being created using the data model of the database (Galea,

Figure 3 and Column7 Lines 23-41, where it is clear that a data model or GUI database

302 is received by the modeler 304) and teaches the limitations:

a) "receiving a request from the user interface from a client" (Galea, Column 5,

Lines 51-53, i.e., *a user accesses a server through client 106, 108 via wide area*

network 112 in order to access items for sale (domains) in the product and GUI

database);

b) "(classifying) the plurality of entities and entity types" (Galea, Column 5 Lines

23-50, i.e., *In one embodiment, the databases contain compiled product or server*

components (domains). Each domain contains descriptive information which bind the

domain elements to their respective GUI representations. Prerequisite tags may indicate

*the **GUI element type** such as, for example, single select **list box**, **multi-select list***

***box**, **radio buttons**, **click boxes**, and **input text field**, or images that present the*

selections options for that domain. Optional domain tags determine the run time

behavior of each GUI element such as, for example, whether. or not to display

invalidated options (and subsequently allow single selections overrides), whether or not

user interface pages are presented as sequential pages or as grouped elements,

whether or not pages/groups for each domain are to be displayed and where they are to

be displayed, whether or not the domain is hidden (not included in the user interface),

whether or not the domain is displayed as a read only form element (whereby the

domain selection may only be made through configuration logic), whether or not to

include local or specific domain and domain options representations, whether or not and

which corresponding images are to be displayed with each option selection, whether or

not and in which form an image element is to be updated with the corresponding image,

whether or not and for which corresponding help pages are available for each domain

and option selection, whether or not and in what form a status text to offer when

navigation to each domain is included, and whether or not the domain is required for

final product selection (order inclusion; This disclosure of Galea clearly teaches that

data model contained in the GUI database 302 contains a plurality of entity types, that

is, the database contains compiled product or server components (also referred to as

domains and each of these domains contains descriptive binding domain elements to

their respective GUI representations. Therefore, domains are entities and each entity

describes a type of data structure such as various graphical user elements such as

radio buttons, click boxes, input text fields etc.*);*), the classifying comprising:

> "(determining whether a first entity in the plurality of entities satisfies a first
>
> set of conditions; and

> classifying the first entity as a first entity type upon determining that the
>
> first entity satisfies the first set of conditions)

b) "creating the description of the user interface (based upon the classification of

the plurality of entities)" (Galea, Column 5 Lines 51-59, i.e., *A user accesses a server*

through client 106, 108 via wide area network 112 in order to access items for sale

(domains) in the product and GUI database. Each product or service configuration

domain, together with the user interface tags, is compiled into a secure, binary

compressed file format. After a user accesses the domain through client 106, 108, the

compiled configuration model is downloaded to the browser of client 106, 108. **The user**

interface is dynamically generated based upon the compiled domain tags).

Galea teaches "a plurality of entities" based upon which user interfaces are

created. However, Galea does not explicitly teach "classifying the plurality of entities

into entity types, the classifying comprising: determining whether a first entity in the

plurality of entities satisfies a first of conditions; and classifying the first entity as a first

entity upon determining that the first entity satisfies the first set of conditions; c) creating

the description of the user interface based upon the classification of the plurality of

entities".

On the other hand, Cason teaches also teaches the limitations:

a) "**receiving a request for the user interface from a user**" (Cason, Figure 1

and Column 3 Lines 29-31, i.e., *Referring further to Fig. 1, the specific panel displays*

122 and 124 provide examples of **graphical user interface(GUI) representations of**

the data model entities *102 and 104, respectively;* Cason, Figure 10A i.e.,

INITIALIZATIONS *&* *DEFAULTS* and Column 9 Lines 22-26, i.e., *Referring now to FIG.*

10, a block diagram illustration of a preferred embodiment of the invention is described

from the view of the user at terminal 200. In this embodiment, **initialization and**

defaults *block 202 references online help file 204, screen layouts & bitmaps 206, user*

custom table 217, and start from selection list 210; Note that by initializing the

system/method, the user herein is requesting to obtain a graphical interface

representation of database models 122 and 124);

b) "**after receiving said request, classifying the plurality of entities into**

entity types" (Cason Column 4 Lines 29-57, i.e., *In an ERwin diagram, the entity is*

represented by drawing a box 280 with the name of the entity on top and all of the

attributes of the entity inside the box. The horizontal line 281 in the entity box divides

the attributes into two sets: keys and non-keys (or, data). The area above the line is the

key area 282, and the area below the line is the data area 284. ***The key is the set of***

attributes that identifies an entity*. A key attribute is an attribute which, either by itself,*

or in combination with other key attributes, forms a unique identifier for an entity.

Primary keys are placed above the line in the key area. A non-key attribute is an

attribute which has not been chosen as a key, and is placed in the data area. ***Entity***

instances in an ERwin model are always identified by key attributes. *Key attributes*

are ***chosen*** *from a set of candidate key attributes, which are attributes, or groups of*

attributes, which can be selected as ***primary keys****, and as a candidate key must*

uniquely identify each instance of the entity, and may have no part null, i.e., empty or

missing. In choosing a primary key from the set of candidate key attributes, ***the data***

modeler trys to find an attribute which will not change its value over the life of

each instance of the entity*, is reasonably small, and is not an intelligent key. An*

intelligent key is a key where the structure of the digits in the identifer indicates

groupings, locations, classifications, dates or the like. Candidate keys not selected as

primary keys are designated as alternate keys, and the symbol AKn, where n is a

number, is placed after those attributes which form the ***alternate key****. Alternate keys*

are used to show different indexes the business will use to access the data; Herein, **key**

attributes of database entities "map" to the "entities" of the claimed invention, which

are classified as primary keys, alternate keys and so on); **and "the classifying**

comprising:

 "determining whether a first entity in the plurality of entities satisfies a first

of conditions" (Cason, Column 4 Lines 29-57, i.e., *Key attributes are **chosen** from a*

set of candidate key attributes, which are attributes, or groups of attributes, which can

*be selected as **primary keys**, and as a candidate key must uniquely identify each*

instance of the entity, and may have no part null, i.e., empty or missing; Herein, "a

primary key" is a first entity which satisfies a set of attributes. Note that a primary could

be a combination of attributes); **and**

 "**classifying the first entity as a first entity type upon determining that the**

first entity satisfies the first set of conditions" (Cason, Column 4 Lines 29-57,

Primary Keys are a first entity type and alternate keys and other types of keys are a

second or other type of keys);

 c) "creating the description of the user interface based upon the

classification of the plurality of entities" (Cason, Column 11 Lines 10-31, i.e., *In*

*accordance with the invention, **each user's database** 218 will include one or more*

***description entities** (see, for example, **table definition** 304 and field definition 306,*

*FIG. 11) and each relationship is given a type from a set including **the type***

*"description", as will be more fully described in connection with FIGS. 11-16. **These***

***entities include a description of a particular record** containing information in a*

plurality of leafs, one leaf for each record, necessary to enforce the cardinality rules. In

this embodiment, invoke a relationship control 236 checks the type of the relationship

linking the entity selected to the entity displayed and, as shown at test 240, invokes

page editor 238 if the relationship has type ***"text description".*** ***Page input editor*** *238*

opens at terminal 219*, **allowing the user to enter to the description entity***

instructions *which define the cardinality rules. Page input editor 238 uniquely numbers*

the lines of the leaf being edited thus providing keys to the lines of data in the

description, closes out the window and returns control to record edit interface 220.

Alternatively, the user may activate cancel 250 key 250 to return control to invoke

relationships 236 without saving any changes made to the selected leaf of the

description entity being edited; **Carson** *Column 12 Lines 5-25, i.e., Referring now to*

FIGS. 11-16, the ERwin diagrams describing the preferred embodiment of the invention

will be described. These FIGS. 11-16 include entity-attribute-relationship (EAR) models

of FIGS. 11, 13 and 15 and physical models of FIGS. 12, 14 and 16. FIGS. 11 and 12

describe the database model. FIGS. 13 and 14 describe ***the user interface model.***

FIGS. 15 and 16 combine the database and user interface models. The EAR model

contains the database model which describes the invention. A physical model describes

an actual implementation of an EAR model of the invention for a given database, and

includes table names, field names within tables, and their type and size. The

relationships in a physical model are the same as in the EAR model. ***The user***

interface model describes the look and feel, which includes the "type of control"

to be invoked, the type of window to be displayed, and the default values to be

placed in the fields as the user defines a database model for some aspect of a

business or other organization or process *(such as the database example set forth*

in connection with FIG. 7) and thereafter works with that database to add, delete,

change records or modify the database structure itself; Note herein that "the user

interface model" of Cason maps to the "description of the user interface" of the claimed

invention. Said "user interface model" of Carson is derived based on "keys" which

uniquely identifies database entities from a single record to a table. As such, said "user

interface model" includes "descriptions" (Carson Column 11 Lines 10-31) and are

derived based on keys); **and**

  "distributing the description to said client in order to allow said client

generate elements of said user interface that permits a user to interact with said

database" (Cason, Column 10 Lines 51-67, i.e., *Change record key 226 processes a*

request by the user to edit and change a primary key value. This is a touchy point in

*data bases, **because everything is driven by the primary key(s).** Usually, controls*

are in place to prevent changing such primary keys because there are many pointers to

that key. In accordance with this preferred embodiment of the invention, change record

key 226 initiates a global change, and all pointers to the record identified by the primary

key being changed are accessed and updated. For example, if a primary key is a name,

and person identified by that name marries, then the key would need to be changed. In

those instances where records are stored in user's database 218 in alphabetical, or

some other such order, get next record key 230 and get previous record key 232 are

provided for accessing the next and previous records, respectively; **Cason** Column 11

Lines 1-9, i.e., *In response to the identification of the entity selected, meta data file 216*

is accessed to determine which relationships exist for it. There will be one control

established at terminal 219 for each relationship defined in meta data file 216 for the

selected entity. ***Just as each entity has one or more primary keys, relationships***

have keys that link entities. *A control may be, for example, a push button, popup*

menu, icon, function key, number/enter key--whatever it takes to start an action;

Carson Column 12 Lines 5-25, i.e., ***The user interface model describes the look and***

feel, which includes the type of control to be invoked, the type of window to be

displayed, and the default values to be placed in the fields as the user defines a

database model for some aspect of a business or other organization or process

(such as the database example set forth in connection with FIG. 7) and thereafter works

with that database ***to add, delete, change records or modify the database structure***

itself; Note that any changes to database entities and their respective keys as well as

relationship(s) between database entities (including respective "types of controls") are

stored in the "user Interface mode" of Cason, from which a graphical user interface is

generated which allows the user to add, delete, change records or modify the database

structure). Note Figure 8 of Cason wherein the user terminal is separated from the

database. Also note Column 8 Lines 19-40, wherein "presentation manager" and

"presentation manager code" on the client generates "graphical user interface" which

represent the database.

At the time the invention was made, it would have been obvious to a person of

ordinary skill in the art to add the feature of classifying database entities into a first entity

type if said entities meets a set of conditions, as taught by Cason, to the method of

Galea so that, in the resultant method, database entities would be classified into a first

entity type if said entities meets a set of condition and a description of user interface

would be created based on said classification. One would have been motivated to do so

in order to provide a "modeling system that takes entities, attributes, and relationships

(with cardinality and optionality on data models) and generates a working user interface"

(Cason, Column 8 lines 15-19).

As per claim 24, Official Note is taken that writing descriptions in XML is

notoriously well known in the art. Galea also teaches XML format in Column 8 Lines 25-

67 and Column 9 Lines 1-22.

As per claim 25, Galea teaches the limitation:

"wherein the classifying and creating are performed automatically without human

assistance" (Galea, Column 6 Lines 20-34, i.e., *The distributed client-side e-commerce*

*service system 100 allows for the **automatic generation** of a user interface for a*

*particular product and/or service model "**on-the-fly**" such that a web user is always*

presented with the latest product or service selection choices).

As per claim 26, Galea teaches the limitation:

"before the classifying, obtaining a current data model of the database, the

current data model reflecting any changes to the database up to when the current data

model is obtained, wherein a current description of the user interface is created using

the current data model of the database" (Galea, Column 6 Lines 20-34, i.e., *The*

*distributed client-side e-commerce service system 100 allows for the **automatic***

generation** of a user interface for a particular product and/or service model **"on-the-fly"

*such that a web user is always presented with the **latest** product or service selection*

choices). In Column 10 Lines 44-57, Galea teaches database modeling. From these

teachings, it can be inferred that Galea obtains a current data model of the database.

As per claim 27, Galea teaches the limitation:

"before the classifying, receiving a request from a client that the description be

created, wherein receiving the request triggers the classifying of entities into entity

types" (Galea, Figure 7 and Column 6 Lines 20-34). Galea teaches the client request as

show in Figure 7 as well as described in Column 6 Lines 20-34, in an e-commerce

system that "*allows for the **automatic generation** of a user interface for a particular*

*product and/or service model **"on-the-fly"** such that a web user is always presented*

*with the **latest** product or service selection choices."* It can be inferred that the use and

presentation of the product and service selections of the e-commerce taught by Galea

can be interpreted as a request for creation of a description of a user interface onto a

client system.

Claim 31 is essentially the same as claim 20 except that it set forth the claimed

invention as a computer readable medium rather than a method and rejected for the

same reasons as applied hereinabove.

As per claim 35, Galea in vie of Cason is directed to a method for generating a

user interface that transacts with a database having a data model containing a plurality

of entities (Galea, Figure 3 and Column7 Lines 23-41, where it is clear that a data

model or GUI database 302 is received by the modeler 304) and teaches the limitations:

"receiving a description of the user interface" (Galea, Column 5 Lines 51-59,

i.e., *A user accesses a server through client 106, 108 via wide area network 112 in*

order to access items for sale (domains) in the product and GUI database. Each product

or service configuration domain, together with the user interface tags, is compiled into a

secure, binary compressed file format) "the description based upon classification of the

plurality of entities into entity types" (Galea, Column 5 Lines 23-50, i.e., *In one*

embodiment, the databases contain compiled product or server components (domains).

Each domain contains descriptive information which bind the domain elements to their

*respective GUI representations. Prerequisite tags may indicate the **GUI element type***

*such as, for example, single select **list box**, **multi-select list box**, **radio buttons**, **click***

***boxes**, and **input text field**, or images that present the selections options for that*

domain. Optional domain tags determine the run time behavior of each GUI element

such as, for example, whether. or not to display invalidated options (and subsequently

allow single selections overrides), whether or not user interface pages are presented as

sequential pages or as grouped elements, whether or not pages/groups for each

domain are to be displayed and where they are to be displayed, whether or not the

domain is hidden (not included in the user interface), whether or not the domain is

displayed as a read only form element (whereby the domain selection may only be

made through configuration logic), whether or not to include local or specific domain and

domain options representations, whether or not and which corresponding images are to

be displayed with each option selection, whether or not and in which form an image

element is to be updated with the corresponding image, whether or not and for which

corresponding help pages are available for each domain and option selection, whether

or not and in what form a status text to offer when navigation to each domain is

included, and whether or not the domain is required for final product selection (order

inclusion; This disclosure of Galea clearly teaches that data model contained in the GUI

database 302 contains a plurality of entity types, that is, the database contains compiled

product or server components (also referred to as domains and each of these domains

contains descriptive binding domain elements to their respective GUI representations.

Therefore, domains are entities and each entity describes a type of data structure such

as various graphical user elements such as radio buttons, click boxes, input text fields

etc.*))* **"wherein the classification comprises classification of a first entity as a first**

entity type" (Galea, Column 5 Lines 23-50 in view of Cason Column 4 Lines 29-57

as applied to claim 20 above ; Galea Figure 8; entities (i.e., user elements) are

determined and classified as *"Images", "Navigation Frame" entities, "Dynamic Content*

Frame elements; Galea, Column 5 Lines 23-50, i.e., *Prerequisite tags may indicate the*

GUI element type *such as, for example, single select* **list box,** *multi-select* **list box,**

radio buttons, click boxes, *and* **input text field,** *or images that present the selections*

options for that domain) "upon determination that the first entity satisfies a first set of

conditions" (Galea Figure 8; entities (i.e., user elements) are determined and classified

as "*Images*", "*Navigation Frame*" entities, "*Dynamic Content Frame elements;* Galea,

Column 5 Lines 23-50, i.e., *Prerequisite tags may indicate the **GUI element type** such*

*as, for example, single select **list box**, **multi-select list box**, **radio buttons**, **click***

***boxes**, and **input text field**, or images that present the selections options for that*

domain) ; and

"generating the user interface using the description of the user interface" (Cason,

Column 10 Lines 51-67 in view of Galea, Column 5 Lines 51-59, i.e., *A user accesses a*

server through client 106, 108 via wide area network 112 in order to access items for

sale (domains) in the product and GUI database. Each product or service configuration

domain, together with the user interface tags, is compiled into a secure, binary

compressed file format. After a user accesses the domain through client 106, 108, the

*compiled configuration model is downloaded to the browser of client 106, 108. **The user***

***interface is dynamically generated based upon the compiled domain tags;** Also*

see **Carson** Column 11 Lines 1-9).

Claim 36 is rejected on the same basis as claim 27.

As per claim 37, Galea teaches the limitation:

"before the receiving: sending preferences for the user interface, the preferences

being utilized in creating the description" (Galea, Column 7 Lines 42-47, i.e., *A user*

accesses server 102 via WAN 112 from client 106. When the user makes a selection as

to a domain (for example, an online camera store), the compiled interactive decision

map 310 for the requested domain is downloaded to client 106. In addition, a client

applet is downloaded to client 106. The client applet downloads the interactive decision

map 310 for the particular domain and builds a multi-page graphical user interface (GUI)

map by looping through the interactive decision map 310 and collecting array elements

for each configuration domain; Galea, Column 11 Lines 65-66, i.e., *Window 900 is*

displayed after a user selects a particular domain for viewing; and Galea, Column 12

Lines 4-6, i.e., *After the user accesses a particular domain, the information is*

downloaded to the client from the server and the client applet downloaded from the

server is used to create the display). The domain received from the client by the

application server can be interpreted as the preference selected by the user and it is

clear that the selected domain will be utilized in creating a description of the user

interface as shown in Figure 15.

As per claim 42, Galea in view of Cason is directed to a computer readable

medium storing a computer program which when executed by at least one processor

generates a user interface that transacts with a database having data model containing

a plurality of entities (Galea, Figure 3 and Column7 Lines 23-41, where it is clear that a

data model or GUI database 302 is received by the modeler 304) and teaches the

limitations:

"instructions for receiving a description of the user interface, the description being

based upon classification of the plurality of entities into entity types wherein the

classification comprises classification of a first entity as a first entity type upon

determination that the first entity satisfies a first set of conditions" (Galea, Column 6

Lines 20-34; Column 10 Lines 44-57; Column 5 Lines 51-59; in view of Cason Column 4

Lines 29-57 as applied to claim 20 above); and

"instructions for generating the user interface using the description of the user

interface" (Galea, Column 10 Lines 44-59).

As per claim 43, Galea in view of Cason teaches the limitation:

"wherein the description of the user interface is created using a current data

model of the database, the current data model reflecting any changes to the database

up to when the description is created" (Galea, Column 6 Lines 20-34, i.e., *The*

*distributed client-side e-commerce service system 100 allows for the **automatic***

generation** of a user interface for a particular product and/or service model **"on-the-fly"

*such that a web user is always presented with the **latest** product or service selection*

choices). In Column 10 Lines 44-57, Galea teaches database modeling. From these

teachings, it can be inferred that Galea obtains a current data model of the database.

As per claim 44, Galea in view of Cason teaches the limitation:

"instructions for sending preferences for the user interface, the preferences being

utilized in creating the description" (Galea, Column 7 Lines 42-47, i.e., *A user accesses*

server 102 via WAN 112 from client 106. When the user makes a selection as to a

domain (for example, an online camera store), the compiled interactive decision map

310 for the requested domain is downloaded to client 106. In addition, a client applet is

downloaded to client 106. The client applet downloads the interactive decision map 310

for the particular domain and builds a multi-page graphical user interface (GUI) map by

looping through the interactive decision map 310 and collecting array elements for each

configuration domain; Galea, Column 11 Lines 65-66, i.e., *Window 900 is displayed*

after a user selects a particular domain for viewing; and Galea, Column 12 Lines 4-6,

i.e., *After the user accesses a particular domain, the information is downloaded to the*

client from the server and the client applet downloaded from the server is used to create

the display). The domain received from the client by the application server can be

interpreted as the preference selected by the user and it is clear that the selected

domain will be utilized in creating a description of the user interface as shown in Figure

15.

As per claim 46, Galea in view of Cason is directed to a system and teaches the

limitations:

"a database having a data model and containing a plurality of entities" (Galea,

Figure 3 and Column7 Lines 23-41, where it is clear that a data model or GUI database

302 is received by the modeler 304) ; and

"a server communicatively coupled to the database" (Galea, Column 5 Lines 11-

15, i.e., *Figure 1 is an illustration of one embodiment for a distributed client-side e-*

commerce service system 100. Referring to Figure 1, **server 100** *is coupled to mass*

storage 104. Server 102 and mass storage device 104 are coupled via wide area

network (WAN); and Galea, Figure 3 : 104 (which is mass storage device) which

comprises *product & GUI data 302*) "for creating a description of a user interface that

interacts with the database" (Galea, Column 5 Lines 51-59, i.e., *A user accesses a*

server through client 106, 108 via wide area network 112 in order to access items for

sale (domains) in the product and GUI database. Each product or service configuration

domain, together with the user interface tags, is compiled into a secure, binary

compressed file format. After a user accesses the domain through client 106, 108, the

compiled configuration model is downloaded to the browser of client 106, 108. ***The user***

interface is dynamically generated based upon the compiled domain tags), "the

description being based upon classification of the plurality of entities into entity types"

(Galea Column 5 Lines 23-50 in view of Cason Column 4 Lines 29-57 as applied to

claim 20 above; Galea, Column 5 Lines 23-50, i.e., *In one embodiment, the databases*

contain compiled product or server components (domains). Each domain contains

descriptive information which bind the domain elements to their respective GUI

representations. Prerequisite tags may indicate the ***GUI element type*** *such as, for*

example, single select ***list box***, ***multi-select list box***, ***radio buttons***, ***click boxes***, *and*

input text field, *or images that present the selections options for that domain. Optional*

domain tags determine the run time behavior of each GUI element such as, for

example, whether. or not to display invalidated options (and subsequently allow single

selections overrides), whether or not user interface pages are presented as sequential

pages or as grouped elements, whether or not pages/groups for each domain are to be

displayed and where they are to be displayed, whether or not the domain is hidden (not

included in the user interface), whether or not the domain is displayed as a read only

form element (whereby the domain selection may only be made through configuration

logic), whether or not to include local or specific domain and domain options

representations, whether or not and which corresponding images are to be displayed

with each option selection, whether or not and in which form an image element is to be

updated with the corresponding image, whether or not and for which corresponding help

pages are available for each domain and option selection, whether or not and in what

form a status text to offer when navigation to each domain is included, and whether or

not the domain is required for final product selection (order inclusion; This disclosure of

Galea clearly teaches that data model contained in the GUI database 302 contains a

plurality of entity types, that is, the database contains compiled product or server

components (also referred to as domains and each of these domains contains

descriptive binding domain elements to their respective GUI representations. Therefore,

domains are entities and each entity describes a type of data structure such as various

graphical user elements such as radio buttons, click boxes, input text fields etc.)

"wherein the classification comprises classification of a first entry as a first entity type

upon determination that the first entity satisfies a first set of conditions" (Galea Figure 8;

entities (i.e., user elements) are determined and classified as *"Images", "Navigation*

Frame" entities, "Dynamic Content Frame elements; Galea, Column 5 Lines 23-50, i.e.,

*Prerequisite tags may indicate the **GUI element type** such as, for example, single*

*select **list box**, **multi-select list box**, **radio buttons**, **click boxes**, and **input text field**,*

or images that present the selections options for that domain).

As per claim 47, Galea in view of Cason teaches the limitation:

"wherein the server obtains a current model of the database, the current data model reflecting any changes to the database up to when the current model is obtained, a current description of the user interface being created using the current data model of the database" (Galea, Column 6 Lines 20-34, i.e., *The distributed client-side e-commerce service system 100 allows for the* **automatic generation** *of a user interface for a particular product and/or service model* **"on-the-fly"** *such that a web user is always presented with the* **latest** *product or service selection choices*). In Column 10 Lines 44-57, Galea teaches database modeling. From these teachings, it can be inferred that Galea obtains a current data model of the database.

As Claim 48, Galea teaches the limitation:

"wherein the server is in persistent communication with the database" (Galea, Figure 3).

As per claim 49 Galea in view of Cason teaches the limitations:

"wherein the server is communicatively coupled to a first client via a network" (Galea Figure 1) and "distributes the created description to the first client for enabling the first client to generate the user interface" (Galea, Column 5 Lines 51-59 and Column 6 Lines 20-34).

As per claim 50, Galea teaches the limitation:

"wherein the server provides the first client an only point of access to the database" (Galea, Figure 1 and Figure 3).

As per claim 52, Galea in view of Cason teaches is directed to a computer and teaches the limitations:

a) "a description of a data store" (Galea, Galea, Column 5 Lines 51-59, i.e., *A user accesses a server through client 106, 108 via wide area network 112 in order to access items for sale (domains) in the product and GUI database. Each product or service configuration domain, together with the user interface tags, is compiled into a secure, binary compressed file format. After a user accesses the domain through client 106, 108, the compiled configuration model is downloaded to the browser of client 106, 108. The user interface is dynamically generated based upon the compiled domain tags*)

b) "a browser" (Galea, Figure 6, i.e., *Browser Application*); and

c) "an application for generating user interface elements by using said description and by retrieving a data set from said data store to populate at least one user interface element, said user interface elements for displaying in said browser" (Galea, Galea, Column 5 Lines 51-59 i.e., *A user accesses a server through client 106, 108 via wide area network 112 in order to access items for sale (domains) in the product and GUI database. Each product or service configuration domain, together with the user interface tags, is compiled into a secure, binary compressed file format. After a user accesses the domain through client 106, 108, the compiled configuration model is*

downloaded to the browser of client 106, 108. The user interface is dynamically

generated based upon the compiled domain tags; **Galea** Column 11 line 64 through

Column 12 line 16, i.e., FIG. 9 illustrates an exemplary information block input window

900. Window 900 is displayed after a user selects a particular domain for viewing. The

information used to draw or paint to the HTML page are shown in FIG. 9 are

downloaded from server to client after a user accesses a particular domain. In the

example of FIG. 9, the information is used to display a medicine or drugstore e-

commerce application. After the user accesses a particular domain, the information is

downloaded to the client from the server and the client applet downloaded from the

server is used to create the display. Window 900 includes a number of fields for

selecting specific information contained in the downloaded interactive decision map file.

FIG. 10 illustrates an exemplary interactive decision map window 1000. Decision map

window 1000 is displayed after the user selects specific information from information

block input window 800. Window 1000 includes a number of input and drop selection

boxes and for choosing and selecting information downloaded from server 102) and "at

least one user interface element for receiving queries for the data store" (Galea, Figure

11, i.e., SEARCH 1126 and Galea, Column 12 Lines 16-27, i.e., FIG. 11 illustrates an

exemplary feature selection window 1100. Window 1100 includes a feature selection

pop up window 1102 containing a number of feature selection pull down boxes 1104-

1124. In addition, window 1100 also includes **a search box 1126** and a browse box

1128. After a domain file has been down loaded from the server to the client, the user

may make selections of the various products from any of the pull down boxes (1104-

1124). For example, the user may pull down picture quality box 1104 to give the

selections shown at 1144. In this example, picture qualities may be selected based on

poor, decent, good, very good, excellent or any combination deemed suitable).

As per claim 53, Galea in view of Cason teaches the limitation:

"comprising a storage for storing the description, the browser and the application"

(Galea Figures 1, 3, and 6).

As per claim 56, Galea in view of Cason teaches the limitation:

"wherein said browser is a web browser" (Galea Figure 6), "wherein said

application is a distributed application running on said web browser" (Galea, Figure 3).

As per claim 57, Official Note is taken that distributing an application, as an

applet is notorious well known in the art, as can be seen Java applets on millions of web

sites. Galea also teaches a Java runtime applet on client 106 in column 6 lines 60-62.

As per claim 58, Galea in view of Cason teaches the limitations:

a) "receiving a first request for a first user interface to transact with a first data

store" (Galea Column 5 Lines 51-55, i.e., *A user access a server through client 106, 108*

via wide area network 112 in order to access items for sale (domains) in the product

and GUI database. Each product or service configuration domain, together with the user

interfaces tags , is compiled into a secure, binary compressed file format. And Galea,

Column 5 Lines 10-22, i.e., *FIG. 1 is an illustration of one embodiment for a distributed client-side e-commerce service system 100. Referring to FIG. 1, server 102 is coupled to mass storage device 104. Server 102 and mass storage device 104 are coupled via wide area network (WAN) 112 to a variety of clients 106 and 108. Wide area network 112 may be coupled to any of a variety of clients 106 and 108. In one embodiment, mass storage device 104 contains a product component and graphical user interface (GUI) database and an interactive decision map database. In an alternate embodiment, each of the databases may be contained in a separate mass storage devices 104);*

b) supplying a first description to generate the first user interface" (Galea, Column 4 Lines 1-5, i.e., *the constraint-based configuration file is mapped into a **plurality of pages** at a client, and a page of **the plurality of page** is updated directly from the constraint-based configuration file;* Column 5 Lines 19-22, i.e., *In an alternate embodiment, **each of the databases may be contained in a separate mass storage devices 104;*** Column 5 Lines 23-26, i.e. *In one embodiment, the databases contain compiled product or service components (domains). Each domain contains descriptive information which bind the domain elements to their respective GUI representations;* Figure 12 and Figure 13; Column 12 Lines 28-50, which explains Figure 12 and Figure 13; Also note Figure 1 wherein *Client 106* (first client) and *Client 108* (second client) can access the server for different user interfaces));

c) "receiving a second request for a second user interface to transact with a second data store" (Galea Column 5 Lines 51-55, i.e., *A user access a server through client 106, 108 via wide area network 112 in order to access items for sale (domains) in*

the product and GUI database. Each product or service configuration domain, together

with the user interfaces tags , is compiled into a secure, binary compressed file format.

And Galea, Column 5 Lines 10-22, i.e., *FIG. 1 is an illustration of one embodiment for a*

distributed client-side e-commerce service system 100. Referring to FIG. 1, server 102

is coupled to mass storage device 104. Server 102 and mass storage device 104 are

coupled via wide area network (WAN) 112 to a variety of clients 106 and 108. Wide

area network 112 may be coupled to any of a variety of clients 106 and 108. In one

embodiment, mass storage device 104 contains a product component and graphical

user interface (GUI) database and an interactive decision map database. In an alternate

embodiment, each of the databases may be contained in a separate mass storage

devices 104; Figure 12 and Figure 13; Column 12 Lines 28-50, which explains Figure

12 and Figure 13; Also note Figure 1 wherein *Client 106* (first client) and *Client 108*

(second client) can access the server for different user interfaces); Particular note

Galea's disclosure which states that ***"In an alternative embodiment, each of the***

databases may be contained in a separate mass storage devices 104, which clearly

teaches that there are more than one data stores which different clients (users) could

access;

and d) supplying a second description to generate the second user interface,

wherein said first and second descriptions differ" (Galea, Column 4 Lines 1-5, i.e., *the*

*constraint-based configuration file is mapped into a **plurality of pages** at a client, and a*

*page of **the plurality of page** is updated directly from the constraint-based*

configuration file; Column 5 Lines 19-22, i.e., *In an alternate embodiment, **each of the***

databases may be contained in a separate mass storage devices 104; Column 5

Lines 23-26, i.e. *In one embodiment, the databases contain compiled product or service*

components (domains). Each domain contains descriptive information which bind the

domain elements to their respective GUI representations; Figure 12 and Figure 13;

Column 12 Lines 28-50, which explains Figure 12 and Figure 13; Also note Figure 1

wherein *Client 106* (first client) and *Client 108* (second client) can access the server for

different user interfaces).

As per claim 63 Galea in view Cason teaches the limitation:

"wherein the first and second data stores are the different data stores, wherein

the first request is received from a first user while the second request is received from a

second user different than the first, wherein the description is supplied to the first user

while the second description is supplied to the second user" (Galea, Column 4 Lines 1-

5, i.e., *the constraint-based configuration file is mapped into a **plurality of pages** at a*

*client, and a page of **the plurality of page** is updated directly from the constraint-based*

configuration file; Column 5 Lines 19-22, i.e., *In an alternate embodiment,* **each of the**

databases may be contained in a separate mass storage devices 104; Column 5

Lines 23-26, i.e. *In one embodiment, the databases contain compiled product or service*

components (domains). Each domain contains descriptive information which bind the

domain elements to their respective GUI representations; Figure 12 and Figure 13;

Column 12 Lines 28-50, which explains Figure 12 and Figure 13; Also note Figure 1

wherein *Client 106* (first client) and *Client 108* (second client) can access the server for

different user interfaces).

As per claim 75, Galea in view of Cason teaches the limitation:

"wherein said request from said client initiates the classification of the entities into

entity types" (Cason, Figure 10A i.e., ***INITIALIZATIONS** & DEFAULTS* and Column 9

Lines 22-26, i.e., *Referring now to FIG. 10, a block diagram illustration of a preferred*

embodiment of the invention is described from the view of the user at terminal 200. In

*this embodiment, **initialization and defaults** block 202).*

As per claim 76, Galea in view of Cason teaches the limitation:

"wherein each entity corresponds to at least one table of said database" (Cason,

Figure 1, i.e., 104 (which is a table) and 102 (which is a table)).

As per claim 77, Galea in view of Cason teaches the limitation:

"further comprising a set of instructions for sending said description to said client

in order to allow said client to generate elements of said user interface from said

description" (Galea, Column 5 Lines 51-59, i.e., *A user accesses a server through*

client 106, 108 via wide area network 112 in order to access items for sale (domains) in

the product and GUI database; Cason, Column 10 Lines 51-67; Cason Column 11 lines

1-9), "said user interface elements allowing a user to transact with the database"

(Cason Column 11 lines 1-9, i.e., *"thereafter works with that database **to add, delete,***

change records or modify the database structure itself").

As per claim 78, Galea in view of Cason teaches the limitation:

"wherein an entity describes a data object of said database through properties

comprising at least one of an attribute and a relationship" (Cason, Cason Column 4

Lines 29-57, i.e., *In an ERwin diagram, the entity is represented by drawing a box 280*

with the name of the entity on top and all of the attributes of the entity inside the box.

The horizontal line 281 in the entity box divides the attributes into two sets: keys and

non-keys (or, data). The area above the line is the key area 282, and the area below the

line is the data area 284. ***The key is the set of attributes that identifies an entity;***

Cason *Column 11 Lines 1-9, i.e., In response to the identification of the entity selected,*

meta data file 216 is accessed to determine which relationships exist for it. There will be

one control established at terminal 219 for each relationship defined in meta data file

216 for the selected entity. ***Just as each entity has one or more primary keys,***

relationships have keys that link entities).

As per claim 79, Galea in view of Cason teaches the limitation:

"wherein the entity corresponds to a table of said database" (Cason, Figure 1,

i.e., 104 (which is a table) and 102 (which is a table)) ;

As per claim 80, Galea in view of Cason teaches the limitation:

a) "from a client, receiving a request for the user interface" (Galea, Column 5,

Lines 51-53, i.e., *a user accesses a server through client 106, 108 via wide area*

network 112 in order to access items for sale (domains) in the product and GUI

database; Cason, Figure 1 and Column 3 Lines 29-31, i.e., *Referring further to Fig. 1,*

*the specific panel displays 122 and 124 provide examples of **graphical user***

***interface(GUI) representations of the data model entities** 102 and 104, respectively;*

Cason, Figure 10A i.e., ***INITIALIZATIONS** & DEFAULTS*);

      b) "in response to said request, creating a description of said user interface"

(Galea, Column 5 Lines 51-59, i.e., *A user accesses a server through client 106, 108*

via wide area network 112 in order to access items for sale (domains) in the product

and GUI database. Each product or service configuration domain, together with the user

interface tags, is compiled into a secure, binary compressed file format. After a user

accesses the domain through client 106, 108, the compiled configuration model is

*downloaded to the browser of client 106, 108. **The user interface is dynamically***

generated based upon the compiled domain tags; Cason, Column 11 Lines 10-31,

i.e., *In accordance with the invention, **each user's database** 218 will include one or*

*more **description entities** (see, for example, **table definition** 304 and field definition*

*306, FIG. 11) and each relationship is given a type from a set including **the type***

"description"**, as will be more fully described in connection with FIGS. 11-16. **These

***entities include a description of a particular record** containing information in a*

plurality of leafs, one leaf for each record, necessary to enforce the cardinality rules. In

this embodiment, invoke a relationship control 236 checks the type of the relationship

linking the entity selected to the entity displayed and, as shown at test 240, invokes

*page editor 238 if the relationship has type "text description". **Page input editor** 238*

opens at terminal 219, allowing the user to enter to the description entity

***instructions** which define the cardinality rules. Page input editor 238 uniquely numbers*

the lines of the leaf being edited thus providing keys to the lines of data in the

description, closes out the window and returns control to record edit interface 220.

Alternatively, the user may activate cancel 250 key 250 to return control to invoke

relationships 236 without saving any changes made to the selected leaf of the

description entity being edited; **Carson** *Column 12 Lines 5-25, i.e., Referring now to*

FIGS. 11-16, the ERwin diagrams describing the preferred embodiment of the invention

will be described. These FIGS. 11-16 include entity-attribute-relationship (EAR) models

of FIGS. 11, 13 and 15 and physical models of FIGS. 12, 14 and 16. FIGS. 11 and 12

*describe the database model. FIGS. 13 and 14 describe **the user interface model.***

FIGS. 15 and 16 combine the database and user interface models. The EAR model

contains the database model which describes the invention. A physical model describes

an actual implementation of an EAR model of the invention for a given database, and

includes table names, field names within tables, and their type and size. The

*relationships in a physical model are the same as in the EAR model. **The user***

interface model describes the look and feel, which includes the "type of control"

to be invoked, the type of window to be displayed, and the default values to be

placed in the fields as the user defines a database model for some aspect of a

***business or other organization or process** (such as the database example set forth*

in connection with FIG. 7) and thereafter works with that database to add, delete,

change records or modify the database structure itself; Note herein that "the user

interface model" of Cason maps to the "description of the user interface" of the claimed

invention. Said "user interface model" of Carson is derived based on "keys" which

uniquely identifies database entities from a single record to a table. As such, said "user

interface model" includes "descriptions" (Carson Column 11 Lines 10-31) and are

derived based on keys);

c) "sending said description to said client" (Galea, Column 5 Lines 51-59, i.e *After*

a user accesses the domain through client 106, 108, the compiled configuration model

is downloaded *to the browser of client 106, 108*);

d) "from said client, receiving a request for data to populate at least one user

interface elements of said user interface" (***Galea*** Column 11 line 64 through Column 12

line 16, i.e., FIG. *9 illustrates an exemplary information block input window 900. Window*

900 is displayed after a user selects a particular domain for viewing. The information

used to draw or paint to the HTML page are shown in FIG. 9 are downloaded from

server to client after a user accesses a particular domain. In the example of FIG. 9, the

information is used to display a medicine or drugstore e-commerce application. After the

user accesses a particular domain, the information is downloaded to the client from the

server and the client applet downloaded from the server is used to create the display.

Window 900 includes a number of fields for selecting specific information contained in

the downloaded interactive decision map file.   FIG. 10 illustrates an exemplary

interactive decision map window 1000. Decision map window 1000 is displayed after

the user selects specific information from information block input window 800. Window

1000 includes a number of input and drop selection boxes and for choosing and

selecting information downloaded from server 102);

e) "retrieving said data from a data store" (***Galea*** Column 11 line 64 through

Column 12 line 16, i.e., FIG. *9 illustrates an exemplary information block input window*

900. Window 900 is displayed after a user selects a particular domain for viewing. The

information used to draw or paint to the HTML page are shown in FIG. 9 are

downloaded from server to client after a user accesses a particular domain. In the

example of FIG. 9, the information is used to display a medicine or drugstore e-

commerce application. After the user accesses a particular domain, the information is

downloaded to the client from the server and the client applet downloaded from the

server is used to create the display. Window 900 includes a number of fields for

selecting specific information contained in the downloaded interactive decision map file.

FIG. 10 illustrates an exemplary interactive decision map window 1000. Decision map

window 1000 is displayed after the user selects specific information from information

block input window 800. Window 1000 includes a number of input and drop selection

boxes and for choosing and selecting information downloaded from server 102) and

f) "sending said data to said client in order to allow said client to populate said

user interface element, said client enabled thereupon to generate said user interfaced

which permits a user of said client to transact with said data store" (***Galea*** Column 11

line 64 through Column 12 line 16, i.e., FIG. *9 illustrates an exemplary information block*

input window 900. Window 900 is displayed after a user selects a particular domain for

viewing. The information used to draw or paint to the HTML page are shown in FIG. 9

are downloaded from server to client after a user accesses a particular domain. In the

example of FIG. 9, the information is used to display a medicine or drugstore e-

commerce application. After the user accesses a particular domain, the information is

downloaded to the client from the server and the client applet downloaded from the

server is used to create the display. Window 900 includes a number of fields for

selecting specific information contained in the downloaded interactive decision map file.

FIG. 10 illustrates an exemplary interactive decision map window 1000. Decision map

window 1000 is displayed after the user selects specific information from information

block input window 800. Window 1000 includes a number of input and drop selection

boxes and for choosing and selecting information downloaded from server 102 ; Cason

Column 11 lines 1-9, i.e., *"thereafter works with that database **to add, delete, change***

***records or modify the database structure itself"**).*

As per claim 81, Galea in view of Cason teaches the limitation:

"after sending said data to said client, receiving from the user of said data, a

query for data stored in said data store" (Galea, Figure 11, i.e., *SEARCH 1126* and

Galea, Column 12 Lines 16-27, i.e., *FIG. 11 illustrates an exemplary feature selection*

window 1100. Window 1100 includes a feature selection pop up window 1102

containing a number of feature selection pull down boxes 1104-1124. In addition,

*window 1100 also includes **a search box 1126** and a browse box 1128. After a domain*

file has been down loaded from the server to the client, the user may make selections of

the various products from any of the pull down boxes (1104-1124). For example, the

user may pull down picture quality box 1104 to give the selections shown at 1144. In

this example, picture qualities may be selected based on poor, decent, good, very good,

excellent or any combination deemed suitable).

As per claim 82, Galea in view of Cason teaches the limitation:

"wherein said request for data to populate the user interface element is received

automatically from the client prior to receiving a transaction or a query from the user of

said client" (***Galea*** Column 11 line 64 through Column 12 line 16, i.e., FIG. *9 illustrates*

an exemplary information block input window 900. Window 900 is displayed after a user

selects a particular domain for viewing. The information used to draw or paint to the

HTML page are shown in FIG. 9 are downloaded from server to client after a user

accesses a particular domain. In the example of FIG. 9, the information is used to

display a medicine or drugstore e-commerce application. After the user accesses a

particular domain, the information is downloaded to the client from the server and the

client applet downloaded from the server is used to create the display. Window 900

includes a number of fields for selecting specific information contained in the

downloaded interactive decision map file.   FIG. 10 illustrates an exemplary interactive

decision map window 1000. Decision map window 1000 is displayed after the user

selects specific information from information block input window 800. Window 1000

includes a number of input and drop selection boxes and for choosing and selecting

information downloaded from server 102 ; Cason Column 11 lines 1-9, i.e., "*thereafter*

*works with that database **to add, delete, change records or modify the database***

structure itself").

As per claim 83, Galea in view of Cason teaches the limitation:

"obtaining a data model of said data store" (Galea, Column 6 Lines 20-34, i.e.,

*The distributed client-side e-commerce service system 100 allows for the **automatic***

generation** of a user interface for a particular product and/or service **model "on-the-

***fly"** such that a web user is always presented with the **latest** product or service*

selection choices),

"wherein said creating said description comprises dynamically generating a

description of said user interface based on an analysis of said data model in response

to the request for the user interface from said client" (Galea, Column 6 Lines 20-34, i.e.,

*The distributed client-side e-commerce service system 100 allows for the **automatic***

generation** of a user interface for a particular product and/or service model **"on-the-fly"

*such that a web user is always presented with the **latest** product or service selection*

choices).

As per claim 84, Galea in view of Cason teaches the limitations:

a) "obtaining a data model of a database" (Galea, Column 6 Lines 20-34, i.e.,

*The distributed client-side e-commerce service system 100 allows for the **automatic***

generation** of a user interface for a particular product and/or service model **"on-the-fly"

*such that a web user is always presented with the **latest** product or service selection*

choices);

b) "receiving a request for a user interface from a client" (Galea, Column 5, Lines

51-53, i.e., *a user accesses a server through client 106, 108 via wide area network 112*

in order to access items for sale (domains) in the product and GUI database; Cason,

Figure 1 and Column 3 Lines 29-31, i.e., *Referring further to Fig. 1, the specific panel*

*displays 122 and 124 provide examples of **graphical user interface(GUI)***

***representations of the data model entities** 102 and 104, respectively;* Cason, Figure

10A i.e., ***INITIALIZATIONS** & DEFAULTS*);

c) in response to the request, dynamically creating a description of said user

interface based on an analysis of said data model" (Galea, Column 6 Lines 20-34, i.e.,

*The distributed client-side e-commerce service system 100 allows for the **automatic***

generation** of a user interface for a particular product and/or service model **"on-the-fly"

*such that a web user is always presented with the **latest** product or service selection*

choices; Galea, Column 5 Lines 51-59, i.e., *A user accesses a server through client*

106, 108 via wide area network 112 in order to access items for sale (domains) in the

product and GUI database. Each product or service configuration domain, together with

the user interface tags, is compiled into a secure, binary compressed file format. After a

user accesses the domain through client 106, 108, the compiled configuration model is

*downloaded to the browser of client 106, 108. **The user interface is dynamically***

generated based upon the compiled domain tags; Cason, Column 11 Lines 10-31,

i.e., *In accordance with the invention, **each user's database** 218 will include one or*

*more **description entities** (see, for example, **table definition** 304 and field definition*

*306, FIG. 11) and each relationship is given a type from a set including **the type***

"description", as will be more fully described in connection with FIGS. 11-16. **These**

entities include a description of a particular record containing information in a

plurality of leafs, one leaf for each record, necessary to enforce the cardinality rules. In

this embodiment, invoke a relationship control 236 checks the type of the relationship

linking the entity selected to the entity displayed and, as shown at test 240, invokes

page editor 238 if the relationship has type *"text description"*. **Page input editor** 238

opens at terminal 219, allowing the user to enter to the description entity

instructions which define the cardinality rules. Page input editor 238 uniquely numbers

the lines of the leaf being edited thus providing keys to the lines of data in the

description, closes out the window and returns control to record edit interface 220.

Alternatively, the user may activate cancel 250 key 250 to return control to invoke

relationships 236 without saving any changes made to the selected leaf of the

description entity being edited; **Carson** Column 12 Lines 5-25, i.e., Referring now to

FIGS. 11-16, the ERwin diagrams describing the preferred embodiment of the invention

will be described. These FIGS. 11-16 include entity-attribute-relationship (EAR) models

of FIGS. 11, 13 and 15 and physical models of FIGS. 12, 14 and 16. FIGS. 11 and 12

describe the database model. FIGS. 13 and 14 describe **the user interface model**.

FIGS. 15 and 16 combine the database and user interface models. The EAR model

contains the database model which describes the invention. A physical model describes

an actual implementation of an EAR model of the invention for a given database, and

includes table names, field names within tables, and their type and size. The

relationships in a physical model are the same as in the EAR model. **The user**

interface model describes the look and feel, which includes the "type of control"

to be invoked, the type of window to be displayed, and the default values to be

placed in the fields as the user defines a database model for some aspect of a

business or other organization or process (such as the database example set forth

in connection with FIG. 7) and thereafter works with that database to add, delete,

change records or modify the database structure itself; Note herein that "the user

interface model" of Cason maps to the "description of the user interface" of the claimed

invention. Said "user interface model" of Carson is derived based on "keys" which

uniquely identifies database entities from a single record to a table. As such, said "user

interface model" includes "descriptions" (Carson Column 11 Lines 10-31) and are

derived based on keys);

d) "sending said description to said client in order to allow said client to generate

said user interface from said description, said user interface comprising user interface

elements that permits a user of said client to transact with said database in a manner

conforming to said data model" (*Galea* Column 11 line 64 through Column 12 line 16,

i.e., FIG. *9 illustrates an exemplary information block input window 900. Window 900 is*

displayed after a user selects a particular domain for viewing. The information used to

draw or paint to the HTML page are shown in FIG. 9 are downloaded from server to

client after a user accesses a particular domain. In the example of FIG. 9, the

information is used to display a medicine or drugstore e-commerce application. After the

user accesses a particular domain, the information is downloaded to the client from the

server and the client applet downloaded from the server is used to create the display.

Window 900 includes a number of fields for selecting specific information contained in

the downloaded interactive decision map file.   FIG. 10 illustrates an exemplary

interactive decision map window 1000. Decision map window 1000 is displayed after

the user selects specific information from information block input window 800. Window

1000 includes a number of input and drop selection boxes and for choosing and

selecting information downloaded from server 102 ; Cason Column 11 lines 1-9, i.e.,

"*thereafter works with that database **to add, delete, change records or modify the***

database structure itself");

As per claim 85, Galea in view of Cason teaches the limitation:

"extracting the data model from said database and building the data model from

data that exists in said database" (Galea, Column 5, Lines 51-53, i.e., *a user accesses*

a server through client 106, 108 via wide area network 112 in order to access items for

*sale (domains) in the product and **GUI database**;* Galea, Column 6 Lines 20-34, i.e.,

*The distributed client-side e-commerce service system 100 allows for the **automatic***

generation** of a user interface for a particular product and/or service model **"on-the-fly"

*such that a web user is always presented with the **latest** product or service selection*

choices);

As per claim 86, Galea in view of Cason teaches the limitation:

"wherein said data mode comprises a plurality of entities" (Galea, Column 5

Lines 51-59, i.e., *A user accesses a server through client 106, 108 via wide area*

network 112 in order to access items for sale (domains) in the product and GUI

database. Each product or service configuration domain, together with the user

interface tags, is compiled into a secure, binary compressed file format. After a user

accesses the domain through client 106, 108, the compiled configuration model is

downloaded to the browser of client 106, 108. **The user interface is dynamically**

generated based upon the compiled domain tags; Cason, Column 11 Lines 10-31,

i.e., *In accordance with the invention,* **each user's database** *218 will include one or*

more **description entities** *(see, for example,* **table definition** *304 and field definition*

306, FIG. 11) and each relationship is given a type from a set including **the type**

"description", *as will be more fully described in connection with FIGS. 11-16.* **These**

entities include a description of a particular record *containing information in a*

plurality of leafs, one leaf for each record, necessary to enforce the cardinality rules. In

this embodiment, invoke a relationship control 236 checks the type of the relationship

linking the entity selected to the entity displayed and, as shown at test 240, invokes

page editor 238 if the relationship has type **"text description".** **Page input editor** *238*

opens at terminal 219, **allowing the user to enter to the description entity**

instructions *which define the cardinality rules. Page input editor 238 uniquely numbers*

the lines of the leaf being edited thus providing keys to the lines of data in the

description, closes out the window and returns control to record edit interface 220.

Alternatively, the user may activate cancel 250 key 250 to return control to invoke

relationships 236 without saving any changes made to the selected leaf of the

description entity being edited;) "wherein each entity corresponds to at least one table

of said database" (Cason, Figure 1, i.e., 104 (which is a table) and 102 (which is a

table));

Claim 87 is rejected on the same basis as claim 80.

As per claim 88, Galea in view of Cason teaches the limitation:

"before receiving the data stored in said data store, automatically sending a

request for the data stored in said data store in order to populate the user interface

element" (*Galea* Column 11 line 64 through Column 12 line 16, i.e., FIG. *9 illustrates an*

exemplary information block input window 900. Window 900 is displayed after a user

selects a particular domain for viewing. The information used to draw or paint to the

HTML page are shown in FIG. 9 are downloaded from server to client after a user

accesses a particular domain. In the example of FIG. 9, the information is used to

display a medicine or drugstore e-commerce application. After the user accesses a

particular domain, the information is downloaded to the client from the server and the

client applet downloaded from the server is used to create the display. Window 900

includes a number of fields for selecting specific information contained in the

downloaded interactive decision map file.   FIG. 10 illustrates an exemplary interactive

decision map window 1000. Decision map window 1000 is displayed after the user

selects specific information from information block input window 800. Window 1000

includes a number of input and drop selection boxes and for choosing and selecting

information downloaded from server).

13.   Claims 21 and 22 are rejected under 35 U.S.C. 103(a) as being unpatentable

over Galea in view of Cason and further in view of in view of Roth (U.S. Patent Number

6564218).

As per claim 21, Cason in view of Galea teaches the limitation:

"each entity in the data model describes a type of data object associated

with the database" (Galea, Column 5 Lines 23-50 in view of Cason Column 4 Lines

29-57 as applied to claim 20 above) **and**

"the classifying produces the first entity type for a first group of data

objects and a second entity type of second a second group of data objects, the

data objects in the first group of data objects being updated in the database more

frequently than the data objects in the second groups of data objects" (Cason

Column 4 Lines 29-57).

However, Cason in view of Galea does not explicitly teach the limitation: "**the**

data objects in the first group of data objects being updated in the database more

frequently than the data objects in the second groups of data objects".

On the other hand, Roth teaches the limitation:

"the data objects in the first group of data objects being updated in the database

more frequently than the data objects in the second group of data objects" (Roth,

Column 4 Lines 54-57, i.e., *Advantageously, said one or more supersets to be used for*

said validity check are selected from said specified supersets on the basis of a priori

knowledge of supersets least likely to be updated). Roth teaches supersets of data

objects and some of said supersets are more frequently update while others are less frequently updated.

At the invention was made, it would have been obvious to a person of ordinary skill in the to add the feature of updating data objects/entities, of which some are more frequently updated and some are less frequently updated, as taught by Roth, to the method of Cason in view of Cason so that, in the resultant method, the classifying would produce the first entity type for a first group of data objects and a second entity type for a second group of data objects, the data objects in the first group of data objects being updated in the database more frequently than the data objects in the second group of data objects. One would have been motivated to do so in order to improve overall speed of retrieval/updates of digital information (Roth, Column 4 Lines 39-43).

As per claim 22, Cason in view of Galea and further in view of Roth teaches the limitations:

"wherein the first entity type is a Main entity type and the second entity type is an Enumeration type" (Roth, Column 8 Lines 65 through Column 9 Line 5, i.e., *As **changes typically affect some parameters with a higher probability than others**, the version comparison can be optimized by starting the comparison using the parameter with **the least probability of change, e.g. starting with the parameter "country"**, provided that data are changed on a country by country basis).*

14.   Claim 23, 32, and 51 are rejected under 35 U.S.C. 103(a) as being unpatentable
over Cason in view Galea and further in view of Sexton et al., (hereinafter "Sexton",
(U.S. Patent Number 7093263).

As per claim 23, Cason in view of Galea does not explicitly teach the limitation:
"wherein the description is a generic description configured to be interpreted in different
platforms or operating environments".

On the other hand, Sexton teaches the limitation:

"wherein the description is a generic description configured to be interpreted in
different platforms or operating environments" (Sexton, Column 3 Lines 40-52, i.e., *In
addition, instructions can automatically be generated for getting and setting values in
the object in the platform-independent format, thereby diminishing the reliance on
manually coding the operations. As a result, errors are reduced and the code is more
maintainable).*

At the time the invention was made, it would have been obvious to a person of
ordinary skill in the art to add the feature of employing generic descriptions/instructions
which can be interpreted in different operating environments, as taught by Sexton, to
the method of Cason so that, in the resultant method, the description would be a
generic description which is to be interpreted in different platforms. One would have
been motivated to do so in order to make the code/instructions/descriptions portable,
which is a well-known practice in the art, such as Java code.

Claim 32 is rejected on the same basis as claim 23.

As per claim 51, Galea in view of Cason and further in view of Sexton teaches

the limitations:

"wherein the server is communicatively coupled, via the network, to a second

client" (Galea Figure 1 in view of Sexton Figure 1) "having a different platform or

operating environment than the first client" (Column 9 Lines 3-6, i.e., *For example, if a*

*platform does not have a 4-byte primitive integer (e.g., on 64-bit machines such as **Cray***

YP-1), and "distributes the created description to the second client for enabling the

second client to generate the user interface" (Galea, Column 5 Lines 51-59 and Column

6 Lines 20-34).

15.   Claim 38 is rejected under 35 U.S.C. 103(a) as being unpatentable over Galea in

vie of Cason and further view of Mashayekhi (U.S. Patent Number 5818936).

As per claim 38, Galea in view of Galea does not explicitly teach the limitation:

"before the receiving: sending authentication information".

On the other hand, Mashayekhi teaches the limitation:

"before the receiving: sending authentication information" (Mashayekhi, Column

7 Lines 10-30, i.e., *when a user 201 attempts to access a particular application*

program, such as a local application 240 or network-based application program 236, the

particular application program requires that the user be authenticated prior to accessing

its processes or data).

At the time the invention was made, it would have been obvious to a person of ordinary skill to add the feature of authenticating a user access, as taught by Mashayekhi, to the method of Galea in view of Cason so that the resultant method would comprise sending authentication information. One would have been motivated to do so in order to provide security to the database, which is well known in the art.

16.   Claims 59-62 are rejected under 35 U.S.C. 103(a) as being unpatentable over Galea in view of Cason and further in view of Mashayekhi (U.S. Patent Number 5818936).

As per claim 59, Galea in view of Cason teaches the limitations:

a)"receiving a first request from a first user" (Figure 1 wherein by way of *Client 106* (first client) and *Client 108* (second client), different users can access the server for different user interfaces);

b) "receiving a second request from a second user" (Figure 1 wherein by way of *Client 106* (first client) and *Client 108* (second client), different users can access the server for different user interfaces);

c) "supplying a first description to the first user and a second description to the second user" (Figure 1 wherein by way of *Client 106* (first client) and *Client 108* (second client), different users can access the server for different user interfaces; Galea, Column 4 Lines 1-5, i.e., *the constraint-based configuration file is mapped into a **plurality of pages** at a client, and a page of **the plurality of page** is updated directly from the constraint-based configuration file;* Column 5 Lines 19-22, i.e., *In an alternate*

embodiment, **each of the databases may be contained in a separate mass storage devices 104;** Column 5 Lines 23-26, i.e. *In one embodiment, the databases contain compiled product or service components (domains). Each domain contains descriptive information which bind the domain elements to their respective GUI representations;* Figure 12 and Figure 13; Column 12 Lines 28-50, which explains Figure 12 and Figure 13; Also note Figure 1 wherein *Client 106* (first client) and *Client 108* (second client) can access the server for different user interfaces, wherein the first and second descriptions are different based on the roles of the users).

Galea in view of Cason does not explicitly teach the limitations: "the first and second users having different authentication information".

On the other hand, Mashayekhi teaches the limitation:

"request comprising authentication information" (Mashayekhi, Column 7 Lines 10-30, i.e., *when a user 201 attempts to access a particular application program, such as a local application 240 or network-based application program 236, the particular application program requires that the user be authenticated prior to accessing its processes or data*).

At the time the invention was made, it would have been obvious to a person of ordinary skill to add the feature of authenticating a user access, as taught by Mashayekhi, to the method of Galea in view of Cason so that the resultant method would comprise sending different authentication information by two different user and, because users are different, description sent to each user would be different . One

would have been motivated to do so in order to provide security to the database, which

is well known in the art.

As per claim 60, Galea in view Cason and further in view of Mashayekhi teaches

the limitations:

"wherein the first and second user interfaces comprises at least two interface

elements for facilitating data transactions" (see Galea, Figure 12 and 13, wherein more

than two interface elements are displayed), "wherein said first interface comprises at

least one or more user interface elements than said second user interface" (Galea

Figure 1 describes 2 clients and Galea Figures 12 and 13 describes a plurality of

interface elements on each generated user interfaces. Depending on domains, numbers

of user interface elements are different. As such, one interface always has more

interface elements than other interface).

As per claim 61, Galea in view Cason and further in view of Mashayekhi teaches

the limitation:

"wherein first and second user interfaces are displayed in an application running

different computers" (Galea, Figure 1, *Client 106* (first client) and *Client 108* (second

client).

As per claim 62, Galea in view Cason and further in view of Mashayekhi teaches

the limitation:

"wherein the application is web browser" (Galea, Galea, Column 5 Lines 51-59

i.e., *A user accesses a server through client 106, 108 via wide area network 112 in*

order to access items for sale (domains) in the product and GUI database. Each product

or service configuration domain, together with the user interface tags, is compiled into a

secure, binary compressed file format. After a user accesses the domain through client

*106, 108, the compiled configuration model is **downloaded to the browser of client***

***106**, 108.*).

Contact Information

17.   Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Dennis Myint whose telephone number is (571) 272-

5629. The examiner can normally be reached on 8:30AM-5:30PM Monday-Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, John Breene can be reached on (571) 272-4107. The fax phone number for

the organization where this application or proceeding is assigned is 571-273-5629.

Information regarding the status of an application may be obtained from the

Patent Application Information Retrieval (PAIR) system. Status information for

published applications may be obtained from either Private PAIR or Public PAIR.

Status information for unpublished applications is available through Private PAIR only.

For more information about the PAIR system, see http://pair-direct.uspto.gov. Should

you have questions on access to the Private PAIR system, contact the Electronic

Business Center (EBC) at 866-217-9197 (toll-free).

/dennis myint/

Dennis Myint
Examiner
AU-2162

/John Breene/

Supervisory Patent Examiner, Art Unit 2162